First Hit

Generate Collection Print

L2: Entry 117 of 120 File: TDBD Mar 1, 1990

TDB-ACC-NO: NB9003416

DISCLOSURE TITLE: Technique for Implementing Context Sensitive Text Replacement.

PUBLICATION-DATA:
IBM Technical Disclosure Bulletin, March 1990, US

VOLUME NUMBER: 32
ISSUE NUMBER: 10B
PAGE NUMBER: 416 - 418

PUBLICATION-DATE: March 1, 1990 (19900301)

CROSS REFERENCE: 0018-8689-32-10B-416

DISCLOSURE TEXT:

- This invention detects relational database object name occurrences within a syntactically correct SQL source statement and selectively replaces a (proper) subset of those occurrences with new object names, based upon both the context of the object within the source SQL statement and the type of the object as defined within the local database catalog. - In SQL objects of distinct types (e.g., tables, views, synonyms, columns) share a single name space. Two distinct objects may be known by the same name, and the distinction between the objects can only be deduced from the context of the name within the SQL statement. - For example, a Relational table and one of its constituent columns, as well as a host variable within the application code, may all be named ABC. If this is the case, the distinction between the column, table, host variable, all named ABC, within the following SQL statement must be inherent within the syntactical representation or PARSE of the SQL string: SELECT ABC FROM ABC INTO:ABC; Within a distributed database environment which supports object location transparency, the SQL Application Programming Interface (SQL API) must be strengthened to support access to local objects, which map to access to remote objects. In response to this requirement, an ALIAS object is created locally and provides the local linkage to a remote table or view. The ALIAS object shares the name space of other local database objects and is effectively a pointer to a different object (either a table or a view) which resides at a remote database instance, where it is possibly known by a different name. Prior to processing any SQL statement which accesses objects of type ALIAS, the local DBMS must ensure that the ALIAS represents a table/view at some remote DBMS instance, and substitute the name of that table or view (as it is known at the remote database instance) within the SQL source statement. Since ALIAS objects share the name space with other local database objects, e.g., tables/views, etc., the text replacement must be performed with sensitivity to both object type and object context. - In the example shown above, if ABC is identified as an ALIAS existing at a remote database instance where it is known by the name DEF, then simple text replacement which ignores object type and context would yield the following syntactically correct SQL statement: SELECT DEF FROM DEF INTO :DEF; However, this replacement does not distinguish the context of the ALIAS object from either the column object or the host variable. The correct text replacement would yield: SELECT ABC FROM DEF INTO:ABC; The parse of an SQL statement produces a "parse tree", a syntactical representation of the SQL text. The context and names

of relational objects are known when the SQL statement is parsed. However, the object type is not known until the statement is optimized during BIND processing, which occurs after the parse. - The BIND process traverses the parse tree and utilizes the local database catalog to identify objects of type ALIAS, and for each such ALIAS occurrence, associates the name of an object at some remote database instance at which the object resides. The ALIAS nodes within the parse tree now provide the context for text replacement of the ALIAS object names. That is, the ALIAS nodes, along with the associated mapped object names, identify the correct subset of the source SQL statement which is subject to text replacement. - The invention's technique retains sufficient semantic information during the parse in order to differentiate object types, context and position of the object name within the text string. ALIAS objects and associated mapped names continue to be identified based on the parse tree and the local database catalog. An additional data structure is introduced to capture position of all ALIAS object name occurrences within the SQL text. Utilizing this, the text replacement may be performed by scanning the original text string left-to-right and performing simple text substitution. That is, if all occurrences of object type T named Nm are to be remapped, the algorithm does not remap objects of type C, also named Nm. - The parsing algorithm is enhanced to correlate relative position within the text string with object name and object context. This is performed for all occurrences of objects such that the context may only be of type ALIAS, table, view, synonym or column qualifier. When such an object is recognized, as defined by the SQL grammar, a node within the parse tree is created. - If the parse terminates and no syntax errors are discovered, then the local database catalog is accessed to resolve the types of the objects appearing within the parse tree. A new data structure, the "correlation table" is created whose entries identify nodes within the parse tree representing ALIAS name occurrences (table, view or column qualifier) which must be replaced by the name maintained within the parse tree node. The correlation table entries are sorted by position, and then the original SQL text is positionally scanned, left-to-right, replacing only that text for which an entry exists within the correlation table. By maintaining positional information within the parse tree nodes, the subset of the nodes representing ALIAS object names which require name mapping is available immediately after the object types are differentiated, e.g., ALIAS, column qualifier, etc. - The complexity of the algorithm required to perform this context- sensitive text replacement is linear (O(n)) since it requires a linear traversal of the parse tree in conjunction with a linear (left-to- right) scan of the source SQL statement.

Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

Display: 10 **Documents in Display Format:** - **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search Clear Interrupt

Search History

DATE: Thursday, April 22, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L2	L1 and table	120	L2
L1	(query or sql or "sequential query language")near3 object with view	156	L1

END OF SEARCH HISTORY